

N89 - 16303

524-61
16704.3
5P

WAS-22

DEC Ada* Interface to Screen Management Guidelines(SMG)

by

Somsak Laomanachareon
Dr. Anthony A. Lekkos

University of Houston, Clear Lake

INTRODUCTION

DEC's Screen Management Guidelines are the Run-Time Library procedures that perform terminal-independent screen management functions on a VT100-class terminal. These procedures assist users in designing, composing, and keeping track of complex images on a video screen.

There are three fundamental elements in the screen management model: the pasteboard, the virtual display, and the virtual keyboard. The pasteboard is like a two-dimensional area on which a user places and manipulates screen displays. The virtual display is a rectangular part of the terminal screen to which a program writes data with procedure calls. The virtual keyboard is a logical structure for input operation associated with a physical keyboard. Other features included in SMG are input and output operations, control of asynchronous actions, optimizing performance, and many more.

SMG can be called by all major VAX languages. Through Ada, we use predefined language Pragmas to interface with SMG. They are Pragma Interface and Pragma Import_Valued_Procedure. In association with these Pragmas, we also used the three other predefined packages: System, Condition_Handling, and Starlet. With these predefined Pragmas and packages, we can put together another package that contains all the procedure calls to SMG which allow Ada application programs to access the SMG.

* Ada is a registered trademark of the U.S. Government-Ada Joint Program Office

Supported by NASA/JSC-UHCL Ada Beta-test site contract

The Screen Management procedures provide terminal independence by allowing user to perform all screen functions without concern for the type of terminal being used and if the terminal being used does not support the requested function in hardware, the Screen Management procedures perform the requested function by emulating it in software. The important aspect of the Screen Management Facility is the separation of user programs from the physical device. For example, the user program writes to the virtual display instead of the physical screen. The separation of virtual operations from physical operation allows the terminal-independent aspect to be realized.

Working with the SMG involves three fundamental elements in the screen management model. First, a pasteboard is always associated with a physical device. A pasteboard can be either smaller or larger than the physical screen, but each output device can have only one pasteboard associated with it. A pasteboard can be thought of as a logical coordinate system where position(1,1) corresponds to the upper left-hand corner of the screen. With this coordinate system, the virtual display, described later, can be placed anywhere and it may be partly visible on the physical screen.

Second, a virtual display is a rectangular part of the terminal screen to which a program writes data and text. When a virtual display is associated with a pasteboard, it is said to be pasted. When the display is removed from a pasteboard, it is said to be unpasted. To make a virtual display visible, you have to paste to a pasteboard. Your program can create and maintain several virtual displays and each display can be pasted to more than one pasteboard at the time.

Third, a virtual keyboard is a logical structure for input operation associated with a physical keyboard or it maybe associated with any file accessible through Record Management Services(RMS). The advantage of using virtual keyboards is device independence. The Screen Management procedures maps the different of code sequences into a uniform set of function codes.

All the attributes associated with pasteboards, virtual displays, and virtual keyboards that your program created can be modified and maintained at all times. A virtual display can be pasted, unpasted, and moved around a pasteboard. Input and output of each virtual display is independent of each other.

Text can be added, inserted, and deleted from a virtual display. Their video attributes can also be altered. The cursor position on a virtual display can be requested or set to any position on the virtual display.

The cursor position on a virtual display should not be confused with the physical cursor position on the screen. Although each virtual display has an associated virtual cursor position, only the cursor position on the most recent modified virtual display corresponds to a physical cursor. Line drawing capabilities and control of asynchronous events are also provided as well as support of Non-DIGITAL terminals.

SMG can be called by all major VAX languages. In Ada, predefined language Pragma's are used to interface with SMG. Pragma Interface which allows Ada program to call subprogram written in another language. A Pragma Interface has the following form

```
Pragma Interface (language-name, subprogram-name);
```

Together with Pragma Interface, the Pragma Import_Valued_Procedure is specially designed for calling system routines. System routines return status values using the same parameter-passing as Ada uses for returning function results. Some system routines also cause side effects on its parameters. Ada treats a routine that returns a result as an Ada function, but a function with IN OUT or OUT parameters is not legal in Ada. Pragma Import_Valued_Procedure allows such a routine to be interpreted as a procedure in an Ada program, and as a function in the external environment. Note that the first parameter of the imported procedure must be an OUT parameter passed value. The value is returned as function value. The other parameters can be specified with the mode IN, IN OUT, or OUT, according to the service routine parameters. For example:

```
with System, Condition_Handling;
```

```
package SMG is
```

```
  procedure Create_Pasteboard
    (Status : out Condition_Handling.Cond_Value_Type;
     Pasteboard_Id : out Integer;
     Output_Device : String := String'Null_Parameter;
     Rows, Columns : Integer := Integer'Null_Parameter;
     Screen_Flag : Boolean := Boolean'Null_Parameter);
```

```

pragma Interface (SMG, Create_Pasteboard);
pragma Import_Valued_Procedure
  (Create_Pasteboard, "SMG$CREATE_PASTEBOARD");

procedure Create_Virtual_Display
  (Status : out Condition_Handling.Cond_Value_Type;
   Rows, Columns : Integer;
   Display_Id : out Integer;
   Display_Attribute,
   Video_Attribute,
   Char_Set : System.Unsigned_Longword
    := System.Unsigned_Longword'Null_Parameter);
pragma Interface (SMG, Create_Virtual_Display);
pragma Import_Valued_Procedure
  (Create_Virtual_Display, "SMG$CREATE_VIRTUAL_DISPLAY");

      ...      ...      ...      ...
      ...      Other procedures      ...
      ...      ...      ...      ...

end SMG;

```

From the example above, the package System provides types and operations for manipulating system-related variables and parameters. The package Condition_Handling provides VAX Ada types for VAX/VMS condition values as in the above status parameter which is returned by a system routine. Another package, not shown, is Starlet which provides VAX Ada type, VAX Ada constants for symbol definitions, and VAX Ada operations for calling system and RMS services. The package Starlet is specially useful in the application program which calls procedures in the SMG package that use symbol definition, for example:

```

with SMG, System, Condition_Handling, Starlet;

procedure Screen is
  Status : Condition_Handling.Cond_Value_Type;
  Screen_1 : Integer;
  ...      ...      ...
  ...      ...      ...
begin
  ...      ...      ...
  ...      ...      ...
  SMG.Create_Virtual_Display
    (Status,
     Rows      => 7,
     Columns   => 70,
     Display_Id => Screen_1,
     Video_Attribute => Starlet.SMG_M_REVERSE);

```

```
...      ...      ...  
...      ...      ...  
end Screen;
```

As shown in the example, all output in the virtual display named Screen_1 will be in the reverse video.

With these packages and pragmas, we can put together a package which contains all the Screen Management procedures that we need. Then Ada application programs can use this Screen management package to create and manage application screens.